[<< Back to Main]

Part 6: Adding An AI-enabled Feature

This assignment is a continuation of your progress from Assignment 5. Your task is to enhance the web application you've developed by integrating a novel AI-enabled feature. This assignment encourages innovation and technical skill, blending web development with cutting-edge AI capabilities.

This assignment is an open canvas for you to explore the possibilities of AI in enhancing user experiences and creating more intelligent, responsive, and capable software applications. We look forward to seeing your creative solutions and the practical implementation of AI in your projects. Good luck!

Introduction

The world of web applications is ever-evolving, with AI increasingly becoming an integral part. By adding an AI-enabled feature to your web application, you will not only enhance the user experience but also deepen your understanding of how AI can be integrated into real-world software solutions. This assignment is your opportunity to explore the creative and practical application of AI in web development.

Requirements

- **Innovative Feature**: Integrate a novel feature that significantly enhances the user experience or functionality of your web application.
- **AI Integration**: Incorporate a trained AI model as a core component of the new feature. The AI should serve a clear, beneficial purpose within the application context.
- **REST API Exposure**: The AI functionality should be accessible via a REST API on the backend, ensuring seamless interaction with the web application frontend.
- **Web Technologies**: Unlike the previous assignments, here you are not restricted to the given web technologies and are free to integrate any web technology that you believe will significantly enhance the user experience and functionality of the application. Be innovative and consider how different technologies might interact or complement each other to create a more compelling application.
- **Documentation**: Ensure to document your AI/technology choices and the rationale behind them. Your exploration and justification of chosen technologies will be a valuable part of your learning and assessment.

AI Model Considerations

- **Pre-trained Models**: Opt for existing trained AI models. Custom training or fine-tuning is not required unless specifically approved by the teaching team.
- **Model Types**: You may use any of the following types of AI models:
  - Pre-trained Machine Learning models
  - Foundation Models
  - Large Language Models (LLMs)
  - Visual Models
  - Audio Models
- **Open Source Preference**: Whenever possible, choose open-source models to avoid additional costs and promote accessibility.
- **Hosting**: Models should ideally be hosted locally (e.g., using GPT4All, ollama, Llama.cpp, Mixral). If using commercial or cloud-based APIs or models, ensure that API keys are shared with teaching assistants for evaluation purposes.

Recommendations

- **Language Flexibility**: The AI feature's REST API can be implemented in any programming language. We recommend using the one most familiar to you or best

suited to the chosen AI technology.

- **Python for AI**: Python is highly recommended due to its extensive support and libraries for AI and machine learning.
- **LangChain**: Consider using LangChain for developing applications powered by language models, especially for its comprehensive features and ease of use in integrating complex AI functionalities.

Deliverables

- **Approved Proposal**: Submit a brief description and justification of the chosen AI feature for approval.
- **Running Demo**: Provide a working demonstration of the feature integrated into the web application.
- **Code Submission**: Ensure your code is well-documented, organized, and tagged by the deadline.
- **Documentation**: Include comprehensive documentation covering:
  - A user manual for the new feature.
  - A detailed description of the design and implementation.
    - Include an explanation of the chosen AI model, alternatives considered, the reason for its adoption, its purpose, and its integration.
  - A README doc explaining in detail how to set up and run the code, including
    - any dependencies required
    - any particular versions of depenecies (for example Tensorflow, version 11.22.33)
    - API Keys (if needed to run)

Marking

This part has 35 points in total:

- **Requirements and deliverables**: meeting the requirements and deliverables.
- **Runnable demo**: runnable demo with clear descriptions for running the application.
- **Creativity**: Unique features not implemented by other teams will be highly regarded.
- **Effort**: The amount of work and thoughtfulness put into the feature implementation (merely using an existing AI service with a simple API call will not suffice).
- **Multimodal integration**: Features that effectively utilize various AI modalities (e.g., combining text, visual, and audio) will receive additional recognition.

Submission instructions:

Copy the **commit hash** from Github and enter it in Canvas.

For step-by-step instructions, refer to the tutorial.

Deadline:

These deadlines will be strictly enforced by the assignment submission system (Canvas).

- See Deadlines for dates and the weight (%) of this assignment on the total assignment mark.